CHAPTER VI

TERMINAL WEIGHTED ARRAY GRAMMERS

6.1. INTRODUCTION

Krithivasan and Das [11] have introduced the notion of a terminal weighted grammar, motivated by the idea of describing parquet deformations that are patterns of tiles that shift gradually in one dimension. In a terminal weighted grammar, the terminal generated at any step of a derivation is defined as a function of time. Also utilizing the classes of picture grammars that have been introduced and investigated in [22,23], namely, matrix grammars [22] and array grammars [23] describing rectangular arrays of symbols, terminal weighted matrix grammars have been considered in [11], for describing parquet deformations, namely, patterns changing slowly in two directions. Terminal weighted array grammars have also been considered in [11].

On the other hand motivated by the fact that hexagonal arrays occur in studies on picture processing and scene analysis and the fact that hexagonal arrays can be considered as two-dimensional representations of three-dimensional blocks, grammatical models, called hexagonal array grammars, for generating hexagonal arrays have been introduced in [21,30]. It is natural to introduce the terminal weighted feature in hexagonal array grammars. In this chapter we indicate the terminal weighted feature on hexagonal array grammars.
6.2 TERMINAL WEIGHTED ARRAY GRAMMARS

We recall the notation of terminal weighted one and two-dimensional grammars [11].

We first review (one-dimensional) regular weighted grammar [11].

**Definition 6.2.1** A terminal weighted regular grammar (TWRG) is a 2-tuple \((G, F)\) where \(G\) is a regular grammar \((N, T, P, S)\) and if \(T = \{a_1, \ldots, a_k\}\), \(F\) is a set of \(k\) functions \(\{f_{a_1}, \ldots, f_{a_k}\}\) where \(f_a\) is function from \(P \rightarrow D\), where \(P\) is the set of positive integers and \(D\) is a suitably defined codomain.

**Example 6.2.1** Let \((G, F)\) be a TWRG, where \(G\) is given by \(S \rightarrow aS, S \rightarrow a\), \(F = \{f_a\}\). \(f_a(i) = \) letter L drawn to fit exactly in a square of side \(2^{i-1} \times 2^{i-1}\) units. One unit may be taken suitably. A typical element of \(L(G)\) is given in Fig. 6.1

![Fig. 6.1 A typical element of Example 6.2.1](image)

We next recall terminal weighted array grammars. We refer to [22,23] for notions of array grammars.

**Definition 6.2.2** A terminal weighted array grammar is defined as a 2-tuple \((G,F)\), where \(G\) is an array grammar and \(F = \{f_{a_1}, \ldots, f_{a_k}\}\) where \(f_{a_i}\) is a function \(P \rightarrow D_i, D_1, \ldots, D_k\) being similarly defined codomains.
In a derivation of the array grammar, a string of intermediates is derived with parentheses and row and column catenation operators. The intermediates are replaced starting from the innermost parentheses. Each intermediate is replaced by a suitable element from the intermediate language satisfying the restrictions induced by the row and column catenation operators. The intermediate replaced at the $i^{th}$ step will introduced terminals $a_j$ which are replaced by $f_{a_j}(i)$. If the resultant array formed is not rectangular, it is discarded.

**Example 6.2.2** The following (CF:R) AG generates squares of size $(2n+1) \times 2n+1$,

\[
\begin{align*}
G &= (V, I, P, S) \\
V &= V_1 \cup V_2 \\
V_1 &= \{ s \} \\
V_2 &= \{ A, B, C \} \\
I &= \{ a \} \\
P &= \{ S \rightarrow A \} (B \mid S \mid B) \oplus A), S \rightarrow (c) \} \\
M_C &= \{ a \}, M_A = \{ \frac{a^{2n+1}}{n \geq 1} \}, M_B = \{ \frac{a^{2n+1}}{n \geq 1} \} \\
\text{Let } F &= \{ f_a \} \\
\text{where } f_a(1) &= \text{ a square of side } d \\
f_a(2n) = f_a(2n+1) &= \text{ a square of side } d \\
\text{with } n \text{ concentric square of sides } \frac{d}{n+1}, \frac{2d}{n+1}, \ldots, \frac{nd}{n+1}\]
A typical element of the language generated is given in Fig. 6.2

Fig. 6.2 A typical element of example 6.2.2

6.3 Terminal weighted Hexagonal Array grammars

Hexagonal array grammars, for generating hexagonal arrays have been introduced in [21,30]. We recall these notions.

We consider a triangular grid made up of lines equally inclined and parallel to three fixed directions (upper right , upper left , down ) and their duals (lower left , lower right , up ). We now define the catenation of an a hexagonal array (A) with a b-hexagonal array (H).
If \( H = \)

\[
\begin{array}{cccc}
  & a_{11n} & a & a_{12n} \\
 a_{11n} & a & a_{12n} & a_{1mn} \\
a_{11} & a & a_{12} & a_{2mn} \\
a_{211} & a_{211} & a_{2m1} & a_{2mn} \\
a_{p11} & a_{p11} & a_{pm1} & \cdots \\
  & a_{pm1} & \cdots & \cdots \\
\end{array}
\]

\[
\begin{array}{cccc}
  b_{11s} & b_{12n} & b_{1mn} \\
 b_{111} & b_{122} & b_{1mn} & b_{2rs} \\
b_{11} & b_{12} & b_{2rs} & b_{irs} \\
b_{211} & b_{211} & b_{2r1} & b_{irs} \\
b_{p11} & b_{p11} & b_{pm1} & \cdots \\
  & b_{pm1} & \cdots & \cdots \\
\end{array}
\]

\( A = \)
then $H \otimes A$ is defined if $m = r$ and $p = t$ and is given by

\[
\begin{array}{cccccc}
& b_{11s} & & b_{12s} & & \\
& & b_{112} & & & \\
& & b_{121} & & & \\
& a_{11n} & b_{111} & & A_{1m} & \\
& a_{122} & a_{121} & & a_{1m2} & \\
a_{112} & a_{111} & & & b_{1m1} & \\
& a_{211} & a_{212} & & & \\
& & a_{1m2} & a_{1mn} & & \\
& & a_{1m1} & & & \\
& & & b_{pms} & & \\
& & & a_{pmn} & & \\
& & & a_{pm1} & & \\
& & & & & \\
& a_{pm1} & & & & \\
\end{array}
\]

Similarly, Catenation in the other directions can be defined.

**Definition 6.3.1**

A hexagonal array grammar $G$ is a 5-tuple $(V,I,P,S,L)$ where $V = V_1UV_2$, $V_1$ is a finite set of nonterminals, and $V_2$ is a finite set of intermediates; $I$ is a finite set of terminals; $P = P_1UP_2$ where $P_1$ is a finite set of nonterminal rules of the form $S \rightarrow S_1 \otimes a$ (upper right catenation), $S \rightarrow S_1 \otimes b$ (upper left catenation), $S \rightarrow S_1 \otimes c$ (down catenation) were $S, S_1 \in V_1$, $a,b,c, \in V_2$ and $\otimes$ is upper right arrowhead catenation. $\otimes$ is upper left arrowhead catenation, and $\otimes$ is down arrowhead catenation. $P_2$ is a terminal rule of the forms $S \rightarrow H$ where $S \in V_1$ and $H$ is a hexagonal array over $I$, the
three fixed directions being equally inclined; S is the start symbol; \( L \) is a set of intermediate languages corresponding to each one of the \( k \) intermediates in \( V_2 \).

These Intermediate languages are regular, CF or CS string languages written in the appropriate arrowhead form. An arrowhead is written in the form \( \langle \cdots (v) \cdots \rangle \) which \( \langle v \rangle \) denotes the vertex and the arrowhead is written in the clockwise direction. A hexagonal kolam array grammar is called \((R:R)\), \((R:CF)\), \((R:CS)\) according as all the members of \( L \) are regular, or at least one of \( L \) is CF, or at least one of \( L \) is CS.

By allowing the rules in \( P_1 \) to be of the form \( S \rightarrow (S_1 \leftarrow a) \rightarrow b \), \( S \rightarrow (S_1 \leftarrow a) \rightarrow b \) we get a linear grammar. It is called \((L:R)\), \((L:CF)\), \((L:CS)\) according as all the intermediate languages are regular, atleast one is CF, or atleast one is CS, respectively. We note that the notation is meaningful when we write catenation in the form of left linear rules. In essence, the rule \( S \rightarrow (S \leftleftarrow a) \rightarrow b \) is a linear rule of the form \( S \rightarrow aSb \).

Derivations proceed as follows. For the first stage of derivation, rules in \( P_1 \) and \( P_2 \) are applied sequentially (introducing parentheses along with the arrow-head direction) until all the nonterminals are replaced. For the second stage of derivation, starting from the innermost parentheses, each intermediate is replaced in parallel by an arrowhead of the intermediate language; the length of the arrowhead is determined by condition for arrowhead catenation.

We next give an example to illustrate a \((R:CF)\) hexagonal kolam array grammar.
Example 6.3.1 [30]

Let \( G = (V, I, P, L, S) \) where \( V = V_1 \cup V_2, V_1 = \{S, S_1\}, V_2 = \{a, b, c, x, y, z\} \), \( I = \{G, Y\} \) (where \( G \) stands for gray and \( Y \) for yellow), \( P = P_1 \cup P_2, P_1 = \{S \rightarrow (((S_1 \rightarrow a) \rightarrow b) \rightarrow c), S_1 \rightarrow (((S \rightarrow x) \rightarrow y) \rightarrow z)\} \),

\[
P_2 = \{S \rightarrow H, H = G \}
\]

\[
L = \{L_a, L_b, L_c, L_x, L_y, L_z\}
\]

with

\[
L_a = \{G^n(G)G^n\}, \quad L_b = \{G^n(G)(G^{n+1})\}, \quad L_c = \{G^{n+1}(G)G^{n+1}\},
\]

\[
L_x = \{Y^n(Y)Y^n\}, \quad L_y = \{Y^n(Y)Y^{n+1}\}, \quad L_z = \{Y^{n+1}(Y)Y^{n+1}\},
\]

and \( S \) is the start symbol. Then \( G \) generates the class of hexagonal arrays the first two members of which are given in Fig.6.3.

```
G
G
G
G
Y
Y
G
Y
G
G
Y
G
Y
G
Y
G
Y
G
G
Y
G
G
G
G
G
```

**Fig. 6.3. Hexagonal arrays of Example 6.3.2**
Here we consider the terminal weighted feature in hexagonal array grammars. The concept is analogous to definition 6.2.2. We only illustrate with an example.

Example 6.3.2

We consider the same Hexagonal array grammar in example 6.3.1

A derivation in this grammar is as follows

\[
S \Rightarrow (((s_1 a) \Rightarrow b) \Rightarrow c)
\]

\[
\Rightarrow ((((( s x) y) z) a) \Rightarrow b) \Rightarrow c)
\]

\[
\Rightarrow ((((( H x) y) z) a) \Rightarrow b) \Rightarrow c)
\]

we now define functions, on setting $Y = G$.

\[
f_G(1) = \text{ a circle of diameter } d
\]

\[
f_G(2n+1) = f_G(2n) = \text{ a circle of diameter } d \text{ with } n \text{ concentric circles of diameters}
\]

\[
\frac{d}{n+1}, \frac{2d}{n+1}, \ldots, \frac{nd}{n+1}
\]
A typical element obtained is shown in Fig. 6.4.

![Fig. 6.4 A typical elements of example 6.3.2.](image)

It is clear that the hexagonal array grammars cannot generate these types of pictures if they are not endowed with the terminal weight feature.

### 6.4 CONCLUSION

We have considered terminal weighted hexagonal array grammars to bring out the terminal weighted feature in hexagonal array generation.